

FILE BACKUP METHOD

Publication number: JP11120057 (A)

Publication date: 1999-04-30

Inventor(s): UTSUNOMIYA NAOKI; SONODA KOJI; KUMAZAKI HIROYUKI

Applicant(s): HITACHI LTD

Classification:

- **international:** G06F12/00; G06F12/00; (IPC1-7): G06F12/00

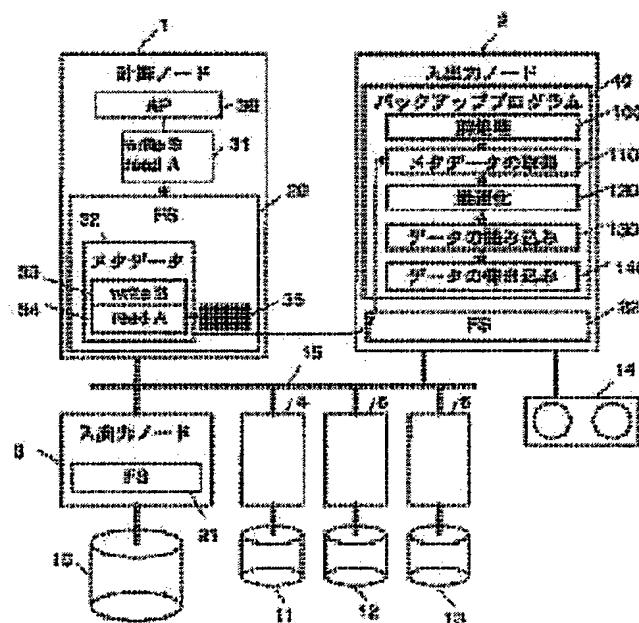
- **European:**

Application number: JP19970284904 19971017

Priority number(s): JP19970284904 19971017

Abstract of JP 11120057 (A)

PROBLEM TO BE SOLVED: To perform a backup process and file access, file by file, at the same time by decreasing the amount of file log information for backup. **SOLUTION:** A file management program 20 records an input/output request 31 from an application program as metadata 32 and a backup program 40 acquires the lock of a file when backup is started and performs a backup process according to the metadata 32 obtained from a file management program 20. After the backup process, the file is unlocked. The file management program 20 prepares a new disk area for the locked file at an update request from a program other than the backup program 40 to prevent it from being overwritten, thereby performing control so that the consistency of the object file of the backup process is maintained.



Data supplied from the **esp@cenet** database — Worldwide

(51)Int.Cl.⁶
G 0 6 F 12/00識別記号
5 3 1F I
G 0 6 F 12/00

5 3 1 M

審査請求 未請求 請求項の数6 O.L (全12頁)

(21)出願番号 特願平9-284904

(22)出願日 平成9年(1997)10月17日

(71)出願人 000005108
株式会社日立製作所
東京都千代田区神田駿河台四丁目6番地
(72)発明者 宇都宮 直樹
神奈川県川崎市麻生区王禅寺1099番地株式会社日立製作所システム開発研究所内
(72)発明者 藤田 浩二
神奈川県川崎市麻生区王禅寺1099番地株式会社日立製作所システム開発研究所内
(72)発明者 熊崎 裕之
神奈川県横浜市戸塚区戸塚町5030番地株式会社日立製作所ソフトウェア開発本部内
(74)代理人 弁理士 小川 勝男

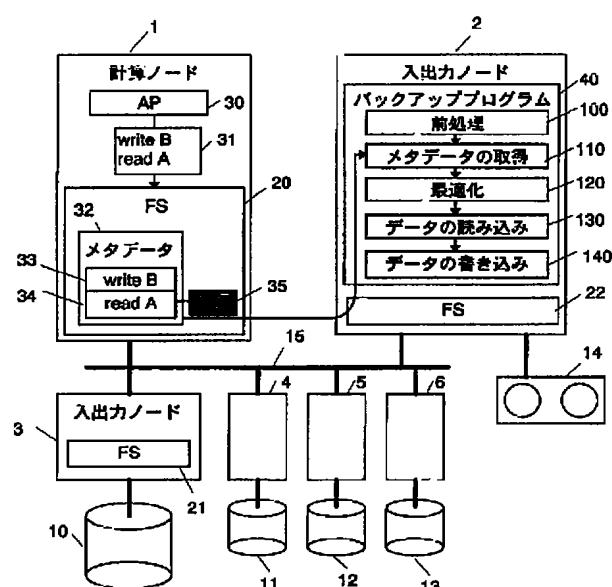
(54)【発明の名称】 ファイルバックアップ方法

(57)【要約】

【課題】バックアップ用ファイルログ情報量を減らし、ファイル単位のバックアップ処理とファイルアクセスの同時実行を可能にする。

【解決手段】ファイル管理プログラム20はアプリケーションプログラムからの入出力要求31をメタデータ32として記録し、バックアッププログラムは、バックアップの開始時にファイルのロックを取得し、ファイル管理プログラムから取得したメタデータに基づいてバックアップ処理を行う。バックアップ処理終了後はロックを開放する。ファイル管理プログラム20はロックのかかったファイルに対して、バックアッププログラム以外から更新要求があった場合は、新たなディスク領域を用意して、上書きを防ぎ、バックアップするファイルの整合性を崩さないように制御する。

図1



【特許請求の範囲】

【請求項1】ディスク装置とテープ装置が接続する複数の出入力ノードと、当該装置が接続しない複数の計算ノードが、専用高速ネットワークで接続した並列計算機システム上で、ファイル単位のバックアップを行うバックアッププログラムと、当該プログラムからのファイルアクセス要求に応答して、出入力を制御するファイル管理プログラムによるファイルバックアップ方法において、ファイルは複数のディスク装置に分散格納され、ファイル管理プログラムは、当該ファイルの更新時に、更新情報をメタデータとして記録し、バックアッププログラムは、指定された当該ファイルのバックアップが最初である場合は、全ファイルをバックアップし、そうでない場合は、ファイル管理プログラムから更新情報メタデータを取得し、更新情報の最適化を行い、更新されたファイル上の領域についてのみバックアップを行い、バックアップ終了後更新情報メタデータを消去することを特徴とするファイルバックアップ方法。

【請求項2】請求項1のファイルバックアップ方法において、バックアップを行う前にファイルに読み書きロックをかけて更新記録に仕切りを入れ、バックアッププログラムがバックアップを行っている最中に、更新要求があり、当該更新要求がバックアップを行う領域と重なっている場合は、上記更新要求を更新情報メタデータに残すとともに、更新要求用データを新たなディスクブロックに記録し、バックアップの終了時、メタデータは仕切りより前を消去し、新たなディスクブロック上の更新データをマージすることを特徴とするファイルバックアップ方法。

【請求項3】請求項2のファイルバックアップ方法において、ミラーされているファイルに対して、バックアップ最中に他プログラムによる更新要求があった場合、ミラー用の複数の複製から一つを選びバックアップ用複製とし、バックアップ要求処理はバックアップ用複製を用いて行い、当該更新要求はバックアップ用複製以外の複製に反映し、バックアップ終了時にバックアップ用複製にそれ以外の複製の内容を反映することを特徴とするファイルバックアップ方法。

【請求項4】請求項1～3のファイルバックアップ方法において、起動されたバックアッププログラムはバックアップ処理を行う前に、物理装置の構成情報を取得し、指定されたバックアップ出力装置が直接接続するノードにバックアッププログラムが移動することを特徴とするファイルバックアップ方法。

【請求項5】請求項1～4のファイルバックアップ方法において、バックアップ管理プログラムがあるノード上で動作し、ファイル管理プログラムは、ある一定量メタデータが蓄積された時点でバックアップサーバに対してバックアップ要求を送り、バックアップサーバは、該当要求にしたがってバックアッププログラムを自動的に起

動しバックアップを行うことを特徴とするファイルバックアップ方法。

【請求項6】請求項5のファイルバックアップ方法において、ファイル管理プログラムは、ある一定量メタデータが蓄積された時点で別のディスクに対して、バックアップを行うことにより、チックポイントデータを自動的に作成することを特徴とするファイルバックアップ方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数のディスク装置上に分散配置されたファイルに対する、並列入出力制御方法に係わり、特にファイルをバックアップする場合、ファイルバックアップ処理とファイルアクセス処理の同時実行を可能とするバックアップ方法に関する。

【0002】

【従来の技術】ディスク装置の故障などによるファイルの破壊を防ぐために、定期的にファイルシステム全体をバックアップする方法は広く利用されている。常にファイルシステム全体をバックアップすることも可能ではあるが、通常運用上は、差分バックアップ方法を利用する。差分バックアップ方法とは、初回に全ファイルシステムのバックアップを行い、次回からは、変更のあったファイルのみをバックアップする方法である。近年、ファイル入出力高速化のニーズや、それに答える並列ファイルシステムの登場により、1ファイルで扱うデータ量は2GBを超える規模となってきた。ファイルシステム全体のバックアップと同様に、單一ファイルのバックアップ処理も膨大な時間を要する。従って、ファイルシステム全体をバックアップする機能以上に、ファイル単位で差分バックアップする機能が重要となってきている。

【0003】UNIXファイルシステムの場合、ファイルシステム全体をバックアップするdumpコマンドなど、ファイルシステムの差分バックアップや、1つのファイルをバックアップするプログラムは存在する。しかし、ファイル単位で差分バックアップを行う機能はない。また、特願平6-306268、「ロギングによるファイル復旧方法及び装置」では、差分データを用いてファイルの復旧を行うが、差分データとして、個々の要求に対する入出力データを全て保持する必要がある。この方法では、途中の入出力データをすべてログとして残す必要があり、入出力単位が大きくなると、容量的に無駄が生じる問題がある。ログストラクチャーファイルシステム (artical entitled "The Design and Implementation of a Log-Structured File System", in ACM Transaction for Computer Systems, Vol. 10, 1992) も同様な問題があ

る。

【0004】

【発明が解決しようとする課題】ファイル単位で差分バックアップを行う場合、更新要求をすべてログ情報（履歴）として保持し、このログ情報に基づいて差分ダンプを行うことは可能である。しかし、ファイル入出力単位が大きい場合、全ての入出力ログ情報量は、元のファイル容量以上に膨大になる可能性がある。また、差分バックアップに用いる場合は、前回バックアップを行った時点のデータとバックアップを行うデータの差分が必要なのであって、途中のログデータ全てが必要ではない。例えば、同じ領域に対してデータAを書き込んだ後に、データBを書き込む場合、始めのデータAは失われてしまつて良いが、ログ方式の場合は、このデータをも保存する。従って、ログ方式における不要ログデータを省く必要がある。

【0005】バックアップ動作中にバックアップするファイルへのアクセスを許可し、書き込みを行うと、データの整合性が崩れてしまう。従って、通常バックアップ動作中は、バックアップを行うファイルやファイルシステムに対して、他のプログラムからのアクセスは禁止である。しかし、ファイルシステムや、ファイルのデータ量の膨大化に伴うバックアップ時間の増大により、バックアップ中のアクセスを禁止することは、運用上問題となる。

【0006】また、通常のバックアップは自動バックアップを行う場合、時間によってバックアッププログラムを起動する。この場合、ファイルに対して更新処理が発生していなかったとしても自動的にバックアップされてしまい、効率的なバックアップを行う上では問題である。

【0007】本発明の目的は、ファイルのログ情報を限定し、従来のログ機能で問題であったログ情報の無駄を省き、そのログ情報を利用するバックアップ方法を提供することにある。

【0008】本発明の他の目的は、ファイル管理プログラムのアクセス制御機能により、データの整合性を崩すことなく、バックアップ動作とファイルアクセス動作の同時実行を可能とする、バックアップ方法を提供することにある。

【0009】本発明のさらに他の目的は、実際のファイル入出力要求量によって自動バックアップを行うことにより、時間による自動バックアップの際に生じる無駄のバックアップ処理を削減することにある。

【0010】

【課題を解決するための手段】本発明は、バックアップを行うファイルに対して、ファイル管理プログラムは、ファイルに対する入出力要求履歴をメタデータとして保存し、バックアッププログラムは、バックアップ処理開始時に、当該要求履歴を取得し、要求履歴に基づいてフ

ァイル上の、差分バックアップを行う領域を特定し、その領域上のデータのみをバックアップする。

【0011】また本発明では、バックアップ処理開始時に、バックアッププログラムはファイル管理プログラムに対してロック取得要求を発行する。ファイル管理プログラムは、ロック中にバックアッププログラム以外のプログラムからの書き込み要求が、バックアップ領域と重なっている場合、新たなディスク領域を割り付けて、そこに書き込み要求を反映する。これにより、バックアップ処理と他プログラムからのアクセスとの同時実行を可能とする。

【0012】また本発明では、ミラーファイルのバックアップ処理時に、ファイル管理プログラムは、ミラーを行う複製のうち一つを選択する。バックアッププログラムからの読み込み要求は、選択した複製に対して処理する。他のプログラムからの読み書き要求がバックアップ中に発生した場合は、選択した複製以外の複製に対して反映する。これにより、バックアップ処理と他プログラムからのアクセスとの同時実行を可能とする。

【0013】また本発明では、蓄積する入出力履歴量に敷居値を設け、この敷居値を超えたときに、ファイル管理プログラムはファイルサーバにバックアップ要求を発行することにより、自動バックアップ処理を行う。これにより、時間による自動バックアップでは、ファイル入出力要求量が少量の時でも発生する、無駄なバックアップ処理を省くことが可能となる。

【0014】

【発明の実施の形態】図1に全体の構成と、バックアッププログラムの簡単な手順を示す。全体は、計算ノード1と入出力ノード3～6が高速ネットワーク15を介して接続する。アプリケーションプログラムはいずれのノード上でも実行可能である。各入出力ノードは物理ディスク装置（10～13）とテープ装置14が接続する。入出力ノード2～6上のアプリケーションプログラムからは、高速ネットワーク15を介さずに直接上記入出力装置にアクセス可能である。例えば、ディスク10は入出力ノード3から、テープ装置14は入出力ノード2から直接アクセス可能である。上記の各ノードと高速ネットワークで1つの計算機システムを構成する。この計算機システムは、アプリケーションプログラムからは、单一の計算機として見える。即ち、アプリケーションプログラムはノードの違いや、接続する入出力装置の接続場所などを意識することなく、各入出力装置にアクセス可能である。全ノードには、ファイル管理プログラム（ファイルサーバ、FSと略記する）が動作し、どのノード上のプログラムからもファイルにアクセスすることが可能となっている。アプリケーションプログラムがファイルにアクセスする場合、このアプリケーションプログラムが動作するノード上のファイル管理プログラム（ローカルファイル管理プログラム）が要求を受け付ける。次

に、ローカルファイル管理プログラムは、他のノード上で動作するファイル管理プログラムと協調して、ファイル入出力処理を実行する。

【0015】ファイル管理プログラムが管理するファイルは、2部分から成る。1つはデータ部で、アプリケーションプログラムが読み書きすることが可能である。もう1つはメタデータで、ファイルの属性を保持する。ファイルの属性は、ファイルサイズ、ファイル作成時刻、アクセスモードなどである。メタデータもディスク装置中に存在して、ファイル管理プログラムはファイル入出力要求に応じてメタデータを更新する。

【0016】入出力ノード2にはバックアップアッププログラムが動作し、テープ装置14に対してファイルのバックアップを行う。計算ノードにはアプリケーションプログラム30(AP)が動作し、ファイル管理プログラム(20)を介してファイルにアクセスする。1ファイルは、複数のディスク装置10～13に分散配置されており、上記複数ディスクへの並列アクセスにより、アプリケーションプログラム30は、高速ファイルアクセスが可能である。

【0017】本実施例では、バックアップアッププログラム40とファイル管理プログラムとが協調して、1ファイルに対する差分バックアップ機能を提供する。バックアッププログラムは出力装置への差分データの書き込み、および、前回のバックアップから新たに更新されたファイル上の領域(更新領域)の特定を行う。ファイル管理プログラムは、バックアップアッププログラムに更新領域特定のために必要な入出力要求履歴を与え、バックアップ時は、バックアッププログラムと他のプログラムの同時アクセスを可能とするための制御を行う。

【0018】次に、アプリケーションプログラムからのファイルアクセスとバックアッププログラムの動作概要を述べる。図1ではアプリケーションプログラムが、ファイル上の領域Aへの読み込みに続き、領域Bへの書き込みを実行している。ファイルサーバ20は、アプリケーションプログラムが送った要求31を、入出力要求履歴33、34とし、これをメタデータ32としてディスク装置10に記録するとともに、ディスク装置への入出力を実行する。このように、ファイルシステムはファイルに対する入出力要求履歴を、メタデータとして記録する。

【0019】バックアッププログラムは、次に述べる手続きにしたがって、テープ装置14にファイルの内容をバックアップする。まず、前処理(100)の後にユーザが指定したファイルに対する入出力要求履歴(メタデータの一部)を取得し(110)、その履歴を解析して最適化を行う(120)。次に、最適化した入出力要求履歴に従って更新された領域のデータのみを読み込み(130)、それをテープ装置へ書き出す(140)。これにより、ユーザが指定するファイルの差分ダンプを

得ることができる。

【0020】以降は、バックアップと同時実行する入出力処理がない場合のバックアッププログラムの処理(単純バックアップ処理)、および、入出力処理とバックアップ処理が同時に動作する場合の処理について詳細に説明する。まず始めに、通常の入出力時、ファイル管理プログラムによるメタデータのロギングについて説明する。

【0021】通常の入出力処理に伴う要求履歴の作成通常の入出力動作に伴う要求履歴作成処理を、図2、図3を用いて説明する。図2は入出力動作に関連するコンポーネントの構成図で、図3はファイル管理プログラム20の入出力動作を示す。ファイル管理プログラムは、アプリケーションプログラムからの要求を受け付ける要求受け付け部(50)と、メタデータを管理するメタデータ管理部(51)、並列入出力を実行する並列入出力実行部(52)から成る。入出力要求履歴に関するメタデータ32は、アプリケーションプログラムの入出力要求に対応して、ファイル領域Aの読み込み要求33と、ファイル領域Bへの書き込み要求34を保持する。メタデータは、通常ファイル管理プログラムがメモリ上にキャッシュし、更新時、定めたディスク装置(メタデータ格納用ディスク装置)に書き込む。ディスク上のメタデータ全体60は固定メタデータ61と動的メタデータ62から成る。固定メタデータのサイズは、メタデータ作成後変化しない。動的メタデータは、時間に伴いサイズが変化する。固定メタデータはファイルのサイズ、ファイルの作成時刻、ファイルのアクセス時刻、アクセスモードなどを保持し、動的メタデータは入出力要求履歴、ストライプ情報を保持する。入出力要求履歴は単調にサイズが増加する。

【0022】次に、入出力時の処理について説明する。アプリケーションプログラム30は始めに、ファイル領域Aからの読み込みを行う。この要求は、アプリケーションプログラム30が動作する計算ノード1上のファイル管理プログラム20が受理する。ファイル管理プログラム20は図3に示すFSの処理手続き(80)に従って、要求を処理する。まず、上述のようにアプリケーションプログラムからの要求を受け取り(81)、この要求に対する要求IDを決定する(82)。要求IDは、アプリケーションプログラムからの要求に対してファイル管理プログラムが与えるIDで、全計算機システムで一意に決まり、しかも、単調増大である。次に、受け取った要求を入出力実行部が実行する(83)。84で実行が成功したかどうかを判断し、成功の場合は、要求IDと共に入出力要求履歴をメタデータ格納用ディスク装置(10)に保存する(85)。入出力要求が失敗した場合は、入出力要求履歴を保存せずに結果を返す。これにより、入出力要求履歴には成功した要求の記録のみが残る。

【0023】入出力要求履歴は、図5に示すレコードとしてメタデータ格納用ディスク装置上の動的メタデータ格納エリアに記録される。各レコードが含むフィールドは、要求ID(90)、要求種別(91)、オフセット(92)、長さ(93)である。例えば、オフセット1024バイトからの128バイトデータの読み込み要求は、95のように要求種別はread、オフセットは1024、長さは128となる。

【0024】単純バックアップ実施例

以下に、バックアッププログラムの動作を説明する。始めにプログラム起動時の動作を説明し、次に、バックアップの手順について説明する。本実施例では、バックアッププログラム動作時、他のプログラムが、ユーザが指定したバックアップファイルにアクセスすることはないと仮定する。

【0025】図4はバックアッププログラム起動時の構成を示す。ユーザ70は入出力ノード7に接続する端末71を介して本計算機システムにアクセスする。ユーザがバックアップ実行を指示すると、バックアッププログラムは、決められたノード上で実行が始まる。どのノード上で実行するかは、計算機システムの運用により定まる。起動したバックアッププログラムは、計算機システムの機器構成情報を取得し、テープ装置が接続する入出力ノードを特定する。バックアッププログラムに対してユーザが出力装置を指定した場合は、その出力装置が存在するノードを特定する。特定が終了すると、バックアッププログラムはOSが提供するリモートプロセス生成機能により、特定したノード上に移動する。

【0026】図4では、バックアッププログラム40は入出力ノード7で動作開始し、テープ装置14が接続する入出力ノード2を特定する。その後、バックアッププログラムは入出力ノード2へ移動し、そこでバックアップ処理を実行する。この処理により、バックアッププログラムが入出力ノード7にあった場合に発生する、ノード7からノード2へのデータ転送オーバヘッドを削除することが可能となる。

【0027】次に、バックアップ処理手順を説明する。ファイルバックアップ処理は、前処理、メタデータの取得、最適化、データの読み込み、データの書き込みの各手順から成る。

【0028】前処理では、ファイル全体をバックアップするかどうかを判断する。ファイル全体のバックアップは、ユーザの指定により行うことも可能である。バックアッププログラムは、書き込みディバイス上の出力媒体のヘッダを参照し、ヘッダ情報が有効でないならば、ファイル全体のバックアップを行う。有効なヘッダであった場合は、差分バックアップを実行する。1ファイルが1つの出力媒体に収まりきらない場合、バックアッププログラムは複数の出力媒体にファイルのデータを出力する。この場合、各出力媒体をボリュームと呼ぶ。

【0029】図6に出力媒体上のデータ(バックアップデータ)フォーマットを示す。全体はヘッダ部とデータ部に分かれる。ヘッダ部は更に、固定部と可変部に分かれる。ヘッダ固定部は、マジック番号200、ファイル情報201、ボリューム情報202、ディレクトリ情報203を有す。マジック番号は、出力装媒体中のデータがバックアップデータであることを示す。バックアッププログラムは、マジック番号のチェックによりヘッダが有効であるかどうかを判断する。ファイル情報は、バックアップを行ったファイルに関する情報を保持し、ファイル名などを含む。ボリューム情報は、この出力媒体固有のデータで、通し番号(ボリューム番号)やヘッダ可変部の長さを含む。例えば、あるファイルが4つのテープにバックアップされた場合、各テープには1から4のボリューム番号が与えられ、これはボリューム情報中に記録される。ディレクトリ情報は、バックアップ履歴を保持する。バックアップ履歴は、ボリューム番号(205)、ボリューム中のオフセット(206)、長さ(207)、日付(208)から成るレコードの集まりである。ボリューム番号とオフセットで、このレコードが表す差分データの格納場所が分かる。例えば、履歴としてボリューム番号3、オフセット295である場合、ボリューム番号3のテープ中、295テープブロックから、差分データが格納されていることを示す。ヘッダ可変部は、バックアップするファイルに固有の情報を含む。例えば、バックアップするファイルが複数ディスク装置上に分散は位置されている場合、ファイル上の領域とその領域が格納されたディスク装置との対応を記録する装置割り付けマップ(図10)の情報を格納する。これにより、上記ファイルのディスク装置上への分散情報が保存できる。データ部には、バックアップした差分データが収まる。特定差分データの媒体上の位置特定は、ファイルのリストア時、ディレクトリ情報を参照して行う。

【0030】前処理ステップ(100)でバックアッププログラムがファイル全体のバックアップを行うと判断した場合は、直ちにデータの書き込みステップ(140)に進む。通常初めてバックアップを行う時が、この場合に相当する。そうでない場合は、メタデータの取得ステップ(110)に進む。

【0031】バックアッププログラムは、メタデータの取得ステップ(120)で、ファイル管理プログラム22に対して要求を発行し、メタデータを取得する(図1参照)。本要求を受理したファイル管理プログラム22は、メタデータがキャッシュ上にあれば、それをバックアッププログラムに返す。キャッシュ上にない場合は、メタデータを管理するノード1上のファイル管理プログラム20からメタデータを受け取り、それをバックアッププログラム40に返す。ファイル管理プログラム20はメタデータをファイル管理プログラム22に転送した後、記録されているメタデータを破棄する。

【0032】最適化ステップ(120)では、取得した入出力要求履歴をスキャンし、不要な要求を削除し、最終的にバックアップ時点で更新された領域を特定する。図8は横軸にファイルのオフセットをとり、縦軸に時間をとったグラフ上に書き込み要求範囲をプロットした図である。要求210が最初の書き込み要求で、要求214が最後の書き込み要求である。例えば、要求214と要求213が重なる領域220は、要求214に従ってファイル管理プログラムが、要求213による書き込みデータ上に新たなデータを上書きする。これにより、要求213の該当領域データは失われる。この2つの要求により更新された領域は、216となる。このようにして、履歴中の全ての要求を調べて、前回のバックアップから今回のバックアップまでに更新された領域215を得る。

【0033】図7は上記最適化を行う手順を示す。入出力要求履歴中には読み込み要求も含まれているので、これを除去する(121)。次に、最新要求から履歴をスキャンして行く。まず始めに、最新要求を変数coverに記憶し(122)、一つ古い要求を変数reqに記憶する(123)。ステップ124でreqがcoverが示す領域に完全に含まれるかどうかを調べる。含まれる場合は、reqが表す要求以降の要求によりreqが示す範囲は上書きされてしまっているので、reqが示す要求を捨てるために、何もせずにステップ126に進む。完全に含まれない場合は、coverが示す領域にマージする(125)。次にスキャンが終了であるかどうかをチェックし(126)、そうであるならば終了する(127)。まだスキャンする要求が残っている場合は、ステップ123からの手順を繰り返す。

【0034】14が最後の書き込み要求である。例えば、要求214と要求213が重なる領域220は、要求214に従ってファイル管理プログラムが、要求213による書き込みデータ上に新たなデータを上書きする。これにより、要求213の該当領域データは失われる。この2つの要求により更新された領域は、216となる。このようにして、履歴中の全ての要求を調べて、前回のバックアップから今回のバックアップまでに更新された領域215を得る。

【0035】図7は上記最適化を行う手順を示す。入出力要求履歴中には読み込み要求も含まれているので、これを除去する(121)。次に、最新要求から履歴をスキャンして行く。まず始めに、最新要求を変数coverに記憶し(122)、一つ古い要求を変数reqに記憶する(123)。ステップ124でreqがcoverが示す領域に完全に含まれるかどうかを調べる。含まれる場合は、reqが表す要求以降の要求によりreqが示す範囲は上書きされてしまっているので、reqが示す要求を捨てるために、何もせずにステップ126に進む。完全に含まれない場合は、coverが示す領域

にマージする(125)。次にスキャンが終了であるかどうかをチェックし(126)、そうであるならば終了する(127)。まだスキャンする要求が残っている場合は、ステップ123からの手順を繰り返す。

【0036】上記の手順を図8の例にあてはめた場合について説明する。まず、coverに要求214を記憶し、reqに、要求213を記憶する。ステップ124でreqはcoverに完全に含まれていないことがわかるので、ステップ125でマージする。ステップ123に戻って、reqに要求212を記憶する。要求212はマージされてできたcoverに含まれていないので、これもcoverにマージする。要求211の場合は、coverに完全に含まれるので、何もしないで次に進む。要求210はマージされて、最終的に215を得る。

【0037】最適化の後、バックアッププログラムは最適化ステップで得られた最適化要求215に基づいて、該当する部分のデータを読み込む(130)。次にデータの書き込みステップ(140)で、データをテープ装置14のテープに書き込む。以上でバックアップが完了する。

【0038】別な実装方法として、最適化ステップに相当する処理を、ファイル管理プログラムが、入出力要求履歴を保存するときに行うこと也可能である。この場合、バックアッププログラムが受理した入出力要求履歴は、既に更新領域を示しているので、バックアッププログラムは最適化手順を省くことができる。

【0039】最後に、リストア処理を簡単に説明する。リストア処理は、リストアを行うユーザが専用のプログラムを起動することにより実行する。図6のディレクトリ情報(203)が示すように、バックアップデータを含む出力媒体は、通常複数の差分データを含んでいる。従って、リストア時はどの時点のデータをリストアするかを指定する必要がある。もし、指定がないときは最新データをリストアする。リストアプログラムは、ヘッダ情報に従って、指定された時点のファイルをリストアする。ディレクトリ部の最初のエントリは一番古い差分データを示す。一番古い差分データには、元のファイルすべてがリストアされている。従って、リストアプログラムは、まずこの差分データを読み込み、ファイルを作成する。次に、指定された時点までの差分データを、上記ファイルに反映することにより、指定された時点でのファイルをリストアする。ヘッダ可変部205に装置割り付けマップが存在する場合は、同マップが示す分散配置情報に基づいてファイルを作成する。但し、正確に指定された物理ディスク装置に配置するか、あるいは、配置形状のみを反映するかはリストアプログラムのオプションである。

【0040】次に、バックアッププログラムとファイルアクセスが同時に動作する場合の処理について説明す

る。

【0041】ファイルアクセスとバックアップの同時実行実施例

単純バックアップ方法は、従来のバックアップ方法と同様に、他プログラムによるバックアップを行うファイル（バックアップファイル）への入出力を許さないバックアップ方法である。従って、何らかの形で他プログラムによる出力要求を抑制する必要があり、適応範囲が限られる。本実施例では、上記の制約を外し、ファイルアクセスとバックアップを同時にを行うことを許すバックアップ方法について述べる。

【0042】図9は、本バックアップ方法の概略を示す図である。単純バックアップ処理との違いは、バックアッププログラムの開始時にバックアップするプログラムに対してロックを取得し、バックアップ処理終了後に同ロックを開放する点である。ファイル管理プログラムは、バックアップ中に発生するバックアップファイルへの更新によって、バックアップデータの整合性が崩れることを防ぐ必要がある。従って、ファイル管理プログラムは更新要求を受け付けると、新たなディスク領域を割り付けて、そこに更新要求を反映する。即ち、ファイル管理プログラムは、この領域に関して一時的に2バージョンを用意し、一方はバックアッププログラム用に元データを残し、他方には、同時アクセスプログラム用に更新データを反映する。以下に、本実施例の場合のバックアッププログラムの動作と、ファイル管理プログラムの動作について述べる。

【0043】バックアッププログラムは、図9に示すように処理の始めて、ファイル管理プログラム（22）に対して、バックアップファイルに対するロックを要求する（150）。このロックの取得により、入出力履歴を固定し、バックアップを行う時点を決定すると共に、これ以降のファイルに対する要求がバックアップ処理に影響しないことをファイル管理プログラムは保証する。以降のバックアッププログラム処理は、前実施例と同じである。但し、メタデータとして取得するのは、ロックをかけた時点の要求履歴である33、34である。要求153、154はロック要求後の入出力要求であるため、バックアッププログラムが処理する履歴情報には含めない。前実施例と同様のバックアップ処理終了後、ステップ151で、ファイルバックアッププログラムはロックの開放をファイル管理プログラムに対して発行し、処理を完了する。

【0044】ファイル管理プログラム22は、本実施例では、入出力履歴情報の固定処理、および、バックアッププログラムからのデータアクセスに対して、同時発生する他のファイル入出力要求が影響を及ぼさないことを保証する処理を行う。バックアップ時点の入出力履歴固定のために、バックアッププログラムが発行したロック要求を受理したファイル管理プログラムは、図11に示

すロック処理（160）を実行する。ステップ161で、ロックするファイルの履歴を更新する可能性のある要求を一時的に停止する。ロック要求受け付け時、ロックするファイルに対する処理が途中の場合は、それを完了させる。ロック要求受け付け以降に到着した新たな要求に対しては、その処理を一時的に停止する。このように、このファイルに対するロック以外の処理を一時中断する。次に、メタデータに仕切り（152）を入れる（162）。仕切りにより、バックアップの対象とする入出力要求履歴を、仕切り以前の履歴に制限する。次に、バックアッププログラムの最適化手順（120）と同様の処理を行い、仕切り以前の履歴によって更新されたファイル上の領域（更新領域）を特定する（163）。最後に、一時停止を解除する。停止解除後ファイル管理プログラムは、当ファイルに関するファイル入出力要求を履歴として保存するが、それは仕切りによって、それ以前の履歴と区別可能である。

【0045】更新領域特定処理（163）の際、ファイル管理プログラムは、装置割り付けマップの分割を行う。装置割り付けマップは、図10に示す構造をしており、ファイル上の領域とその領域のデータを収める物理装置との対応を示す。例えば、図10では、領域170のデータはディスク装置10に格納されることを示す。分割処理では、特定された更新領域と元の装置割り付けマップから、新しい装置割り付けマップ172を作成する。この処理では、装置割り付けマップの各領域について、その領域が更新領域を含んでいれば、更新領域を含む部分と含まない部分に分割する。例えば、領域173（ディスク装置11に割り付けられている）は更新領域を含んでいるので、これを含む領域（174）と含まない領域（175）に分割する。装置割り付けマップ172は、ストライプするディスク装置（10～13）上のファイル管理プログラムに転送される。以上で、ロック処理は終了である。

【0046】次に、バックアップ最中にバックアップを行うファイルへの書き込み要求があった場合の処理について述べる。ファイル管理プログラムは、バックアッププログラム以外のプログラムによる書き込み要求処理時に、装置割り付けマップ172を参照し、書き込み要求領域が更新領域を含んでいる場合、更新領域への上書きをせずに、新たなディスク領域を割り付けてそこにデータを書き込む。ディスク装置10が接続するノード3上のファイル管理プログラム21の処理を図12を用いて説明する。書き込み要求を受理したファイル管理プログラム21は、保持する装置割り付けマップを参照し、ディスク装置10への書き込み要求領域180が更新領域を含むこと特定する。その場合、要求を分割し、更新領域を含まない部分に関しては、上書きする。そうでない部分に関しては、新たなディスク領域184を割り付けて、そこに要求を書き込む。これにより、バックアップ

用のデータは、ディスク領域182、183に存在し、バックアップ中に更新された最新データは、ディスク領域184、183上に存在する。

【0047】領域185に対しては、ディスク上の領域182と184の2つが対応するので、読み込み時は、どちらか一方の領域を選択する必要がある。これは、以下に述べるように、バックアッププログラムと、それと並列動作するアプリケーションプログラムとで、異なるディスク領域にアクセスするように、ファイル管理プログラムが制御する。バックアッププログラムがファイル上の領域170を読み込む場合は、ディスク装置上の領域として181、182、183のデータにアクセスする。これに対して、アプリケーションプログラムが領域170にアクセスする場合は、ディスク上の領域181、184、183にアクセスする。この機能により、バックアッププログラムは、並列動作する他のアプリケーションプログラムによる書き込み要求が発生した場合でも、矛盾なく元のデータにアクセスできる。

【0048】他のディスク装置上のデータに対しても、上記と同様な管理を行う。バックアップ終了時は、バックアッププログラムがロックの開放要求を発行する。これを受け取ったファイル管理プログラムは、バックアップファイルに関するファイル入出力を一時停止し、ディスク領域184のようなバックアップ開始後に更新されたデータを反映する。例えば、ディスク装置10に関しては、ディスク領域184の内容をディスク領域182にコピーする。この場合、データ量が大きい場合は、実際にコピーするのではなく、管理ポインタを張り替えるだけでも良い。次に、仕切りを削除し、一時停止を解除してロック開放処理を終了する。

【0049】自動バックアップ実施例

ここでは、自動バックアップ機能の実施例について説明する。自動バックアップは、通常ある一定の時間間隔でバックアッププログラムを起動する。例えば、OSが一定時間間隔でコマンドを起動する機能を有していた場合、コマンドとしてバックアッププログラムを指定することにより、自動バックアップ機能を実現することができる。この場合、ファイルに対する入出力動作が行われていない場合でも、バックアップ処理が実行され、無駄なバックアップ処理を行う可能性がある。本実施例では、バックアップ処理は一定量のファイルの更新により起動するので、時間による起動に比べて効率的である。

【0050】図13は自動バックアップ時の構成を示す。前記ファイルアクセスとバックアップ同時実行実施例の構成に加え、入出力ノード2にバックアップサーバ190が動作する。自動バックアップ機能を有効にするために、ユーザやシステム管理者の指定によって、ファイル管理プログラムはメタデータ中の入出力要求履歴数居値191とバックアップサーバ情報を設定する。入出力要求履歴量はファイル入出力に従って単調増加するの

で、ある時点でこの量が設定した敷居値を超える。この時、ファイル管理プログラムはバックアップサーバ情報を参照してバックアップサーバを特定し、そのバックアップサーバに対してバックアップ要求192を送信する。バックアップ要求を受信したバックアップサーバ190は、ユーザが起動する場合と同様に、バックアッププログラム40を起動する。バックアッププログラムは、ファイルアクセスとバックアップ同時実行実施例と同様に、バックアップデータをテープ装置14に出力する(193)。一方、アプリケーションプログラム30は、バックアップ処理と同時にファイルに対するアクセスを行うことが可能である。

【0051】さらに、自動バックアップ機能を有効にする時、出力データ先としてファイルを指定することも可能である。これにより、ファイルのチェックポイントデータを自動生成することが可能である。

【0052】ミラーファイルのバックアップ実施例

ここでは、バックアップファイルがミラーファイルである場合の、バックアップ実施例について述べる。ミラーファイルの場合、ファイルは複数の複製を有し、ファイル管理プログラムは、通常、このファイルに対する更新を全ての複製に対して行う。バックアップ時、ファイル管理プログラムは、あらかじめ選択した複製からデータを読み込む。同時実行する他のプログラムからの入出力要求は他の複製に対して実行する。以下に各手順について説明する。

【0053】始めに、バックアッププログラムからのロック取得要求受信時に、ファイル管理プログラムは入出力要求履歴に仕切りを入れた後、複製の選択を行う。複製の選択では、指定されたファイルに関する複数の複製から、一つ複製を選択し、それをバックアップ用複製とし、これ以外の複製を更新用複製とする。次に、ロック処理の完了をバックアッププログラムに通知する。以降、バックアッププログラムからの読み込み要求に対しては、常にバックアップ用複製からデータを読み込む。バックアッププログラムからの書き込み要求はエラーとする。バックアッププログラム以外のプログラムからの要求は、更新用複製に対して要求を反映する。バックアップ処理が完了し、バックアッププログラムがロック開放を要求した時、ファイル管理プログラムは当該ファイルに関する要求を一時停止し、仕切り以降の要求履歴を参照して、更新用複製の更新内容をバックアップ用複製に反映する。次に、ロック開放完了をバックアッププログラムに通知する。以降、このファイルに関する要求は、通常どおりすべての複製に反映する。以上の方法により、バックアップ処理と同時にファイル入出力が可能となる。

【0054】

【発明の効果】本発明によれば、入出力要求の履歴を残し、これを利用することによりファイル単位の差分バッ

クアップが可能である。また、ログデータとして入出力要求のみを収集するので、通常の入出力データも収集するファイルロギング方法に比べて、少容量のログ情報でバックアップ処理を行うことが可能である。また、バックアップ処理とファイルアクセスとの同時実行機能により、バックアップを行いながらファイルにアクセスすることが可能であるので、ファイルのバックアップ処理を待つことなくそのファイルに対する入出力を実行することが可能である。

【図面の簡単な説明】

【図1】実施形態のファイルバックアップ処理が動作する計算機システムの構成図である。

【図2】実施形態のファイル管理プログラムの構成図である。

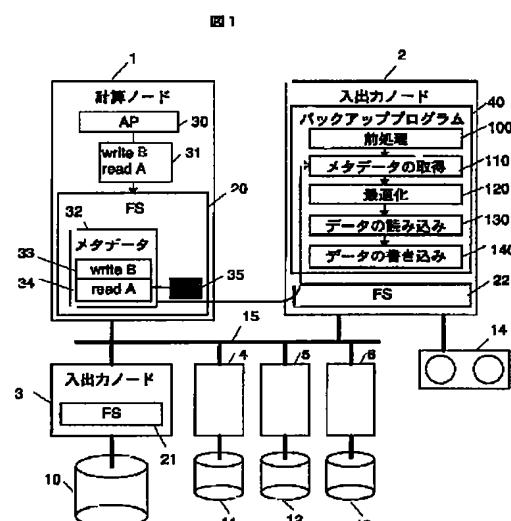
【図3】ファイル管理プログラムの通常の入出力処理の流れを示す図である。

【図4】バックアッププログラム起動時の動作を示す図である。

【図5】メタデータとして記録される入出力要求履歴を示す図である。

【図6】出力媒体上のバックアップデータのフォーマットを示す図である。

【図1】



【図5】

要求 ID	要求種別	オフセット	長さ
90	read	1024	128
91	write	1024	1024
92	write	512	1024
93			

【図7】バックアッププログラムの最適化処理の流れを示す図である。

【図8】更新履歴を時系列的に示す図である。

【図9】ファイルアクセスとバックアップの同時実行を示す図である。

【図10】装置割り付けマップの構成を示す図である。

【図11】ファイル管理プログラムによるロック処理の流れを示す図である。

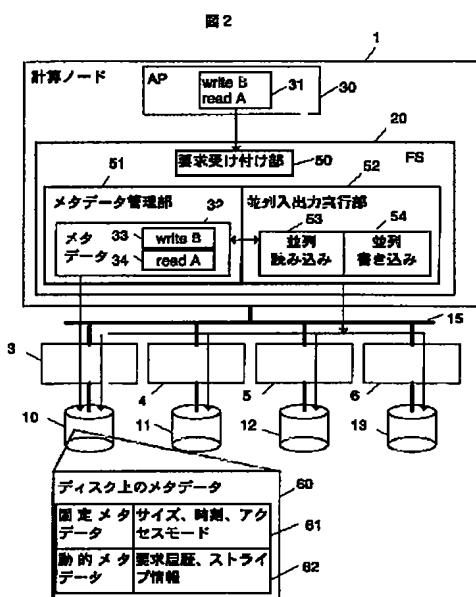
【図12】新たなディスク領域の割り付け処理を示す図である。

【図13】自動バックアップ処理時の構成を示す図である。

【符号の説明】

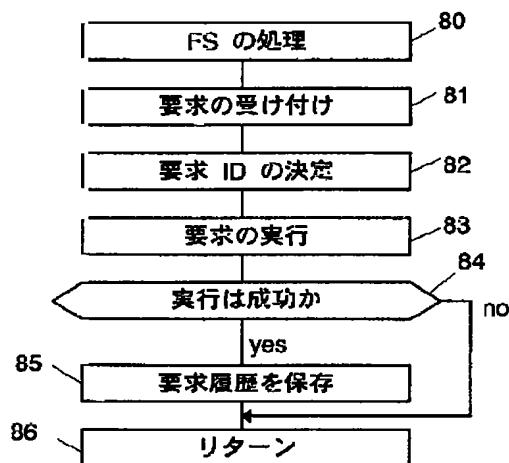
1…計算ノード、2…入出力ノード、30…アプリケーションプログラム、20～22…ファイル管理プログラム、15…高速ネットワーク、10～13…ディスク装置、14…テーブル装置、40…バックアッププログラム、100…前処理ステップ、110…メタデータの取得ステップ、120…最適化ステップ、130…データの読み込みステップ、140…データ書き込みステップ。

【図2】



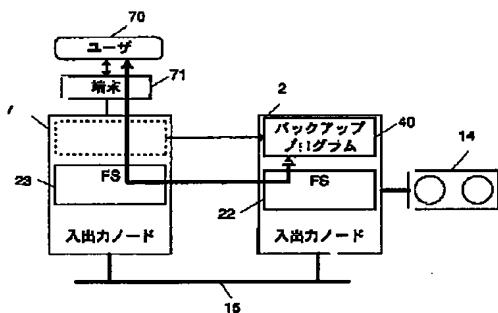
【図3】

図3



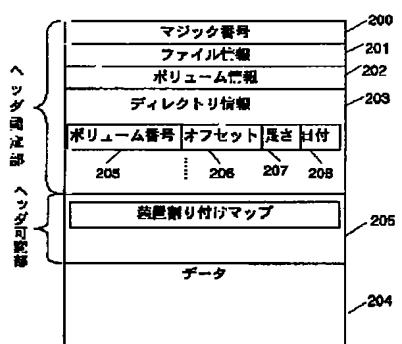
【図4】

図4



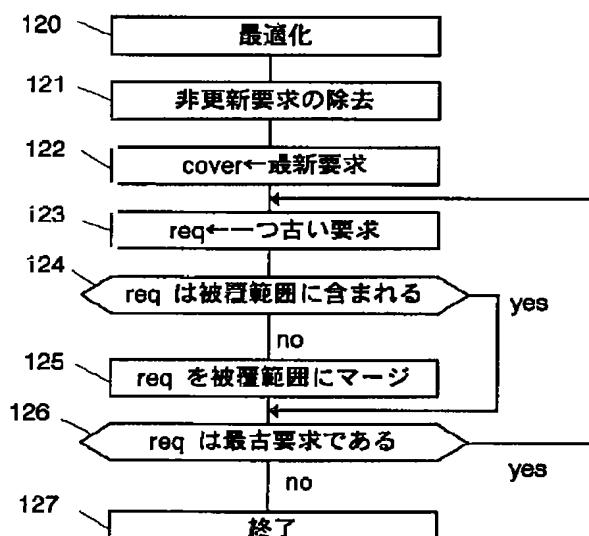
【図6】

図6



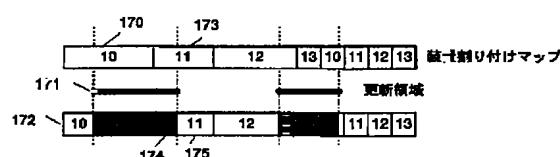
【図7】

図7

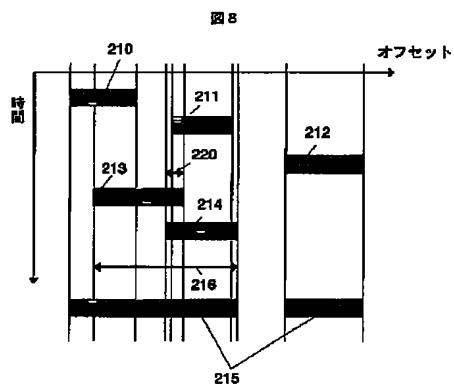


【図10】

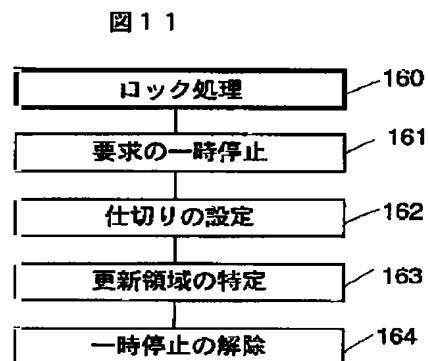
図10



【図8】

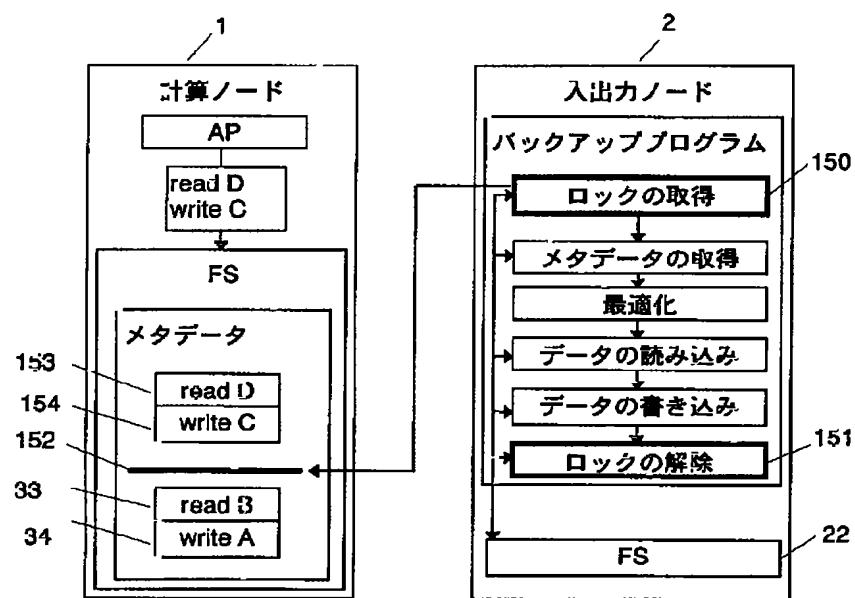


【図11】



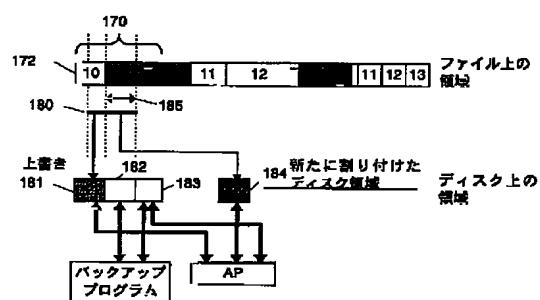
【図9】

図9



【図12】

図12



【図13】

図13

